

Mobile Money is not a Ringtone

Mobile Financial Application Security

Overview

- ✦ Mobile Financial Apps
- ✦ Components
- ✦ Platforms
- ✦ Risks
- ✦ Concerns
- ✦ Recommendations

Based on a presentation for JP Morgan, later for Unnamed Indonesian Telcos... now as hard won hacker knowledge.

Mobile Financial Apps

Applications

- ✦ Mobile Banking Applications
 - ✦ Operator provides channel
- ✦ Mobile Wallet
 - ✦ Operator provides services

So, the FI is either the operator or a third party. This is critical in terms of who assumes the financial risks (typically the FI), and how much control the FI has over the end-to-end security.

Motivators

- ✦ Financial Institutions
 - ✦ Increases stickiness (users configure mobile banking exactly once)
 - ✦ Reduces churn
- ✦ Operators
 - ✦ Increases value of the relationship
 - ✦ Reduces churn

Notice “increases security” isn’t on the list...

Security Goals

- ✦ Authenticate the customer
- ✦ Provide end-to-end security
 - ✦ Confidentiality
 - ✦ Integrity
 - ✦ Availability
- ✦ “At least as secure as an ATM”

Operator Concerns

- ✦ Poor understanding of financial risk management

Operator Concerns, cont.

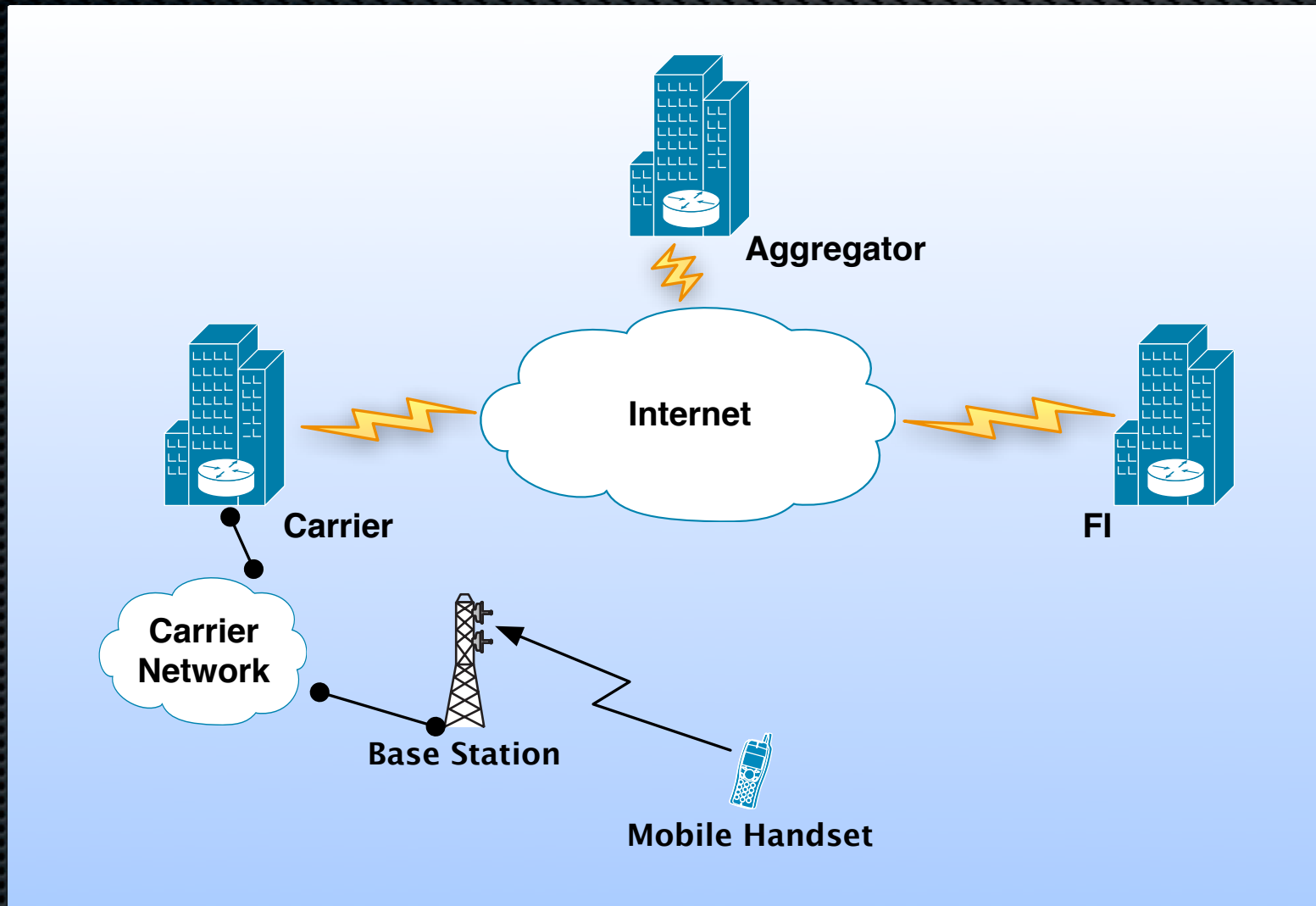
- ✦ Mobile Banking is Value Added Service (VAS)
 - ✦ Ringtones, wallpaper, \$10 tetris clones, all your financial data...
- ✦ Security awareness is limited
 - ✦ Toll fraud: “Will this result in revenue leakage?”

Components

Mobile Elements

- ✦ Handset
- ✦ Over The Air (OTA)
- ✦ Carrier
- ✦ Aggregator
- ✦ Financial Institute (FI)

Elements



Platforms

Handset Applications

- ✦ Web Application
- ✦ “App”
- ✦ SIM Card Application (STK)

Web App

- ✦ Easy to deploy
- ✦ Easy to develop
- ✦ Cross platform
- ✦ Limited control over look and feel
- ✦ Web App Security
 - ✦ SQLi, XSS, CSRF
- ✦ Slow data link
- ✦ Expensive data plans
- ✦ Subset of all phones support browsers

App

- ✦ Complete control over look and feel
- ✦ Powerful operating system
- ✦ Easy to develop*
- ✦ Fractured handset platform landscape
- ✦ Vulnerable to local attacks
- ✦ Hard to secure
 - ✦ Phone developers are not very security aware

As an industry, we don't even know what the mobile handset level threats are... how do we train developers to protect against them?

SIM Application

- ✦ Most secure (potentially)
- ✦ Works on all SIM cards
- ✦ Mature development environment
- ✦ Deployable OTA
- ✦ Secure against malicious phone
- ✦ Cumbersome interface
- ✦ Looks terrible
 - ✦ No multimedia
- ✦ Restricted operating environment
 - ✦ Low power
 - ✦ Low memory

SIM cards are heavily protected to mitigate toll fraud -- mFin apps get this for free Essentially JVM with limited APIs Memory footprint measured in single digit kilobytes... 4k

mBanking Architecture

- ✧ SMS Input
 - ✧ Operator
- ✧ HTTP(S) Input
 - ✧ Aggregator
- ✧ XML Input
 - ✧ Financial Institution

mWallet Architecture

- ✧ SMS Input
 - ✧ Operator
- ✧ HTTP(S) Input
 - ✧ Operator - application
- ✧ Database manipulation

Backend Platforms

- ✦ Problems
 - ✦ Lack of audit trails / logs
- ✦ Single entry book keeping

Risks

Risks

- ✦ Identity
 - ✦ Lost / stolen phone
- ✦ Financial
 - ✦ Fraud
 - ✦ Non-repudiation

More Risks

- ✦ Communications channel
 - ✦ Monitoring / sniffing
 - ✦ Message injection / spoofing
 - ✦ Duplicates

Not Risks

- ✦ Mobile Malware
 - ✦ Not very prevalent
 - ✦ Fractured mobile platform landscape

Mobile Malware has been “next years biggest problem” for at least 2–3 years now.

Attack Opportunities

Handsets

- ✧ Customer identity tied to phone
 - ✧ Lost / stolen == compromised ID
- ✧ Monitoring / Spoofing
- ✧ Malicious client to attack backend systems
- ✧ Infected (not yet...)

OTA

- ✦ Monitoring
 - ✦ GSM encryption is cracked
 - ✦ GSM monitoring equipment for < \$1500

Probably not very practical for scale due to limited range

Operator

- ✦ SMSC based monitoring
 - ✦ SMS processing is unencrypted
- ✦ SMSC based injection
 - ✦ Spoofing SMS from SMSC is trivial

Aggregator

- ✧ Monitoring
 - ✧ Malicious employees
 - ✧ Other customers
- ✧ Injection
 - ✧ See above.

Financial Institutions

- ✦ Poor understanding of Operator concerns

Recommendations

Recommendations

- ✦ Identify customers via a unique mFin PIN + phone
- ✦ Transmit the PIN hashed with the message data
- ✦ Add a unique message ID (timestamp) per customer per request

- ✦ Require customer notification for dangerous operations, e.g. transfers
- ✦ Signup process should include in-branch application
- ✦ Require secure audit trails for all transactions

Anything that modifies accounts

Financial Regulations

- ✦ Require the Carrier to follow financial regulations regarding access and control over the messages
- ✦ Require the Aggregator to follow financial regulations regarding access and control over the messages

- ✦ Use an STK application on the handset
 - ✦ Require code review before it goes live
- ✦ Require security reviews over major components of the environment
 - ✦ Mobile app
 - ✦ Carrier environment
 - ✦ Aggregator environment

these environments are not designed for handling financial data.

- ✦ Develop a clear customer service management plan for lost / stolen handsets
 - ✦ Work with the carrier
 - ✦ Ensure it doesn't automatically cancel CC/ATM

Encryption Keys

- ✦ Manage the encryption keys/certificates used by the application
 - ✦ Work with the Carrier on SIM keys
 - ✦ Work with the Aggregator

Conclusion

- ✦ mFin Apps present unique challenges
- ✦ Trust relationships with third parties
- ✦ Difficult application environments
- ✦ No existing “best practices”
- ✦ Vendors have immature products